

高等学校计算机基础课系列教材

# 计算机 软件基础

辽宁省高校计算机基础课系列教材编委会 组编



大连理工大学出版社 Dalian University of Technology Press

高等学校计算机基础课系列教材

# 计算机软件基础

第四版

辽宁省高校计算机基础课系列教材编委会 组编

主编	李延珩	李振业	朱鸣华
编写	李振业	赵 晶	徐 薇
	李延珩	张丽萍	朱鸣华
	薛伟莲		

大连理工大学出版社

# 辽宁省高校计算机基础课 系列教材编委会

主编 刘百惠

编委 韩玉瑄 李盘林 李振业 张荣存

徐继锋 朱启超 苑致仲 原忠虎 李国义

## 图书在版编目(CIP)数据

计算机软件基础/李延珩,李振业,朱鸣华主编. —4版. —大连:大连理工大学出版社,2002.8

高等学校计算机基础课系列教材

ISBN 7-5611-1354-4

I. 计… I. ①李… ②李… ③朱… II. 软件-高等学校-教材 IV. TP31

中国版本图书馆 CIP 数据核字(2001)第 050337 号

大连理工大学出版社出版发行  
大连市凌水河 邮政编码:116024  
电话:0411-4708842 传真:0411-4701466  
E-mail:dutp@mail.dlptt.ln.cn  
URL:http://www.dutp.com.cn  
大连理工印刷有限公司印刷

---

开本:787毫米×1092毫米 1/16 字数:310千字 印张:13.5

印数:11001—14500册

1998年2月第1版

2002年8月第4版

2002年8月第4次印刷

---

责任编辑:刘新彦

责任校对:耿艳芬

封面设计:孙宝福

---

定价:19.00元

# 前 言

计算机的发展和普及促进了各个学科的相互渗透和发展,引起了现代社会的工作方式、生活方式和思维方式的变化。国家教育部高等学校工科计算机基础教育指导委员会针对非计算机专业的计算机基础教育明确提出了“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的教学模式,本书是根据第二个层次——计算机技术基础的基本要求组织编写的。

全书共分6章:概述,数据结构与算法,操作系统,面向对象程序设计,数据库系统,软件工程基础。为了扩大非计算机专业学生的知识面,使学生了解计算机发展的动态,掌握计算机软件的基本知识,为进一步学习打下良好基础,本书介绍了计算机软件基础知识,尽量作到内容丰富并具有先进性。本书可安排36~56学时,其中讲授24~32学时,上机12~24学时,对于书中内容各学校和专业可根据需要进行取舍,有些章节亦可安排学生自学。为了方便教学和读者学习,本书的部分章节还配有课件,如有需要,请与我们联系。

本书是在作者多年从事非计算机专业教学实践的基础上编写的,具有突出算法的思想性,淡化理论,面向应用及内容新颖等特点。积累多年教学实践,我们对本书内容也进行了多次修订和补充。

本书既可作为本科和大专院校计算机软件技术基础课程的教材,也可作为计算机等级考试的辅导教材、培训教材及广大计算机爱好者的自学用书。

本书由李振业(第1、5章),赵晶、徐薇(第2章),李延珩(第3章),张丽萍(第4章),朱鸣华、薛伟莲(第6章)编写。刘日升、刘润彬、曹桂琴教授对全书进行了认真审读并提出了具体的修改意见,还有一些老师为本书的出版和修订做了许多工作,在此一并感谢。

由于计算机技术发展快,本书内容涉及面广,加之作者水平有限,书中不妥和疏漏之处在所难免,真诚欢迎各位专家和读者批评指正。

编 者

2002年8月

4 5 7 8 10 12

202.118, 74.91

# 目 录

第1章 概述	1
1.1 计算机软件的概念	1
1.1.1 计算机发展简史	1
1.1.2 计算机软、硬件构成	2
1.1.3 计算机软件定义	3
1.2 程序设计语言	4
1.2.1 程序语言的分类	4
1.2.2 语言处理程序	6
1.2.3 典型的高级语言	7
1.2.4 面向对象的语言	8
1.3 网络基础与应用	8
1.3.1 计算机网络的发展	8
1.3.2 计算机网络的分类	10
1.3.3 网络协议与分层模型	11
1.3.4 网络环境与应用	13
习 题	14
第2章 数据结构与算法	15
2.1 概 述	15
2.1.1 数据结构化对数据处理的重要性	15
2.1.2 数据结构研究的三个主要问题	17
2.1.3 算法的基本概念	18
2.1.4 C语言简介	20
2.2 线性表	25
2.2.1 线性表的基本概念	25
2.2.2 线性表的存储结构及其运算	26
2.3 栈和队列	32
2.3.1 栈	32
2.3.2 队列	38
2.4 数组	40
2.4.1 数组的定义	40
2.4.2 数组的顺序存储结构	41
2.4.3 矩阵的压缩存储	41
2.5 树	44
2.5.1 树的定义	44
2.5.2 二叉树	45
2.5.3 哈夫曼树及其应用	50
2.6 图	54
2.6.1 图的基本概念	54

2.6.2	图的存储结构	55
2.6.3	图的遍历	56
<del>2.6.4</del>	<del>最小生成树</del>	<del>57</del>
<del>2.6.5</del>	<del>关键路径</del>	<del>59</del>
2.7	查找	60
2.7.1	顺序查找	60
2.7.2	折半查找	61
2.7.3	分块查找	62
2.7.4	散列查找	63
2.8	排序	65
2.8.1	概述	65
2.8.2	插入排序	65
2.8.3	选择排序	67
2.8.4	交换排序	69
2.8.5	归并排序	71
2.8.6	内部排序方法的选择	72
习题		72

冒泡排序

第3章 操作系统 <sup>3-A</sup> ..... 77

3.1	操作系统导论	77
3.1.1	什么是操作系统	77
3.1.2	操作系统的分类	78
3.1.3	操作系统功能	81
3.1.4	操作系统的特征	82
3.2	几种典型的操作系统	83
3.2.1	DOS系统	83
3.2.2	Windows系列操作系统	85
3.2.3	UNIX系统	89
3.2.4	Linux系统	91
3.3	进程与处理机管理	92
3.3.1	进程的概念及定义	92
3.3.2	进程状态及进程控制块	95
3.3.3	进程控制	98
3.3.4	进程调度	98
3.3.5	进程的同步与互斥	101
3.3.6	进程通信	104
3.3.7	死锁	106
3.4	作业管理与控制	112
3.4.1	用户与操作系统之间的接口	112
3.4.2	作业状态及转换图	113
3.4.3	作业调度	114
3.4.4	作业控制	117
3.5	设备管理	117

重要概念

3.5.1 通道与中断技术	118
3.5.2 缓冲技术	120
3.5.3 设备处理程序	121
3.6 存储管理	121
3.6.1 基本概念	122
3.6.2 存储管理方式	123
3.7 文件管理	129
3.7.1 概述	129
3.7.2 文件的结构与存取方法	130
3.7.3 文件的目录结构	131
3.7.4 文件存储空间的管理	132
3.7.5 文件的共享与文件系统的安全性	134
3.7.6 文件的使用	134
习 题	134
<b>第4章 面向对象程序设计</b>	137
4.1 面向对象概述	137
4.1.1 面向对象思想的由来及技术的发展	137
4.1.2 面向对象程序设计的基本概念	139
4.2 面向对象程序设计基础	142
4.2.1 什么是面向对象的程序设计(OOP)	142
4.2.2 面向过程程序设计与面向对象程序设计的比较	143
4.3 面向对象程序设计语言C++	153
4.3.1 C++对ANSI标准C的扩充	153
4.3.2 C++中的类	153
4.3.3 C++中的构造函数与析构函数	156
4.3.4 C++中的函数重载	157
4.3.5 C++中的继承	158
习 题	160
<b>第5章 数据库系统</b>	162
5.1 数据库的特征与系统构成	162
5.1.1 数据管理技术的发展	162
5.1.2 数据库的特征	163
5.1.3 数据库应用系统	164
5.1.4 数据库的体系结构	165
5.1.5 数据库管理系统	166
5.1.6 存取数据的过程	167
5.1.7 数据库语言	167
5.2 数据模型	168
5.2.1 实体联系模型	168
5.2.2 关系数据模型	171
5.2.3 SQL数据库语言简介	174

5.3 数据库设计 ..... 176

5.3.1 数据库设计的全过程 ..... 176

5.3.2 系统需求分析 ..... 177

5.3.3 概念结构设计 ..... 177

5.3.4 逻辑结构设计 ..... 178

5.3.5 物理结构设计 ..... 179

5.3.6 数据库的实现与维护 ..... 179

5.3.7 数据库的保护 ..... 180

习 题 ..... 182

**第6章 软件工程基础** <sup>1.8半</sup> ..... 184

6.1 软件工程概述 ..... 184

6.1.1 软件的发展与软件危机 ..... 184

6.1.2 软件生命周期 ..... 185

6.1.3 软件工程的基本原则 ..... 186

6.2 软件开发过程基础 ..... 187

6.2.1 软件开发计划的制定与可行性论证 ..... 187

6.2.2 需求分析 ..... 188

6.2.3 概要设计 ..... 189

6.2.4 详细设计 ..... 190

6.2.5 软件编码 ..... 193

6.2.6 软件测试 ..... 194

6.2.7 软件维护 ..... 195

6.3 软件开发中的系统分析与设计方法 ..... 195

6.3.1 结构化系统方法 ..... 196

6.3.2 快速原型法 ..... 199

6.3.3 面向对象方法 ..... 201

6.4 软件重用 ..... 203

6.4.1 重用的概念 ..... 203

6.4.2 软件重用技术 ..... 203

6.5 软件开发的管理技术 ..... 203

6.5.1 软件的质量管理 ..... 204

6.5.2 组织管理 ..... 204

6.5.3 计划与文档管理 ..... 205

习 题 ..... 206

参考文献 ..... 207

模块设计

半页



$\sum_{i=1}^n a_{ij} = 0, ?$   
 $\sum_{j=1}^n a_{ij} = a_{ij} [n]$

# 第 1 章 概 述

## 1.1 计算机软件的概念

### 1.1.1 计算机发展简史

自世界上第一台计算机 ENIAC 问世以来,短短五十多年,计算机技术经历了巨大变革。人们习惯上按计算机硬件技术的水平来划分计算机的发展阶段。

从 1946 年到 20 世纪 50 年代后期为第一代,即电子管时期。计算机的主要逻辑元件是电子管。其特点是体积庞大、功耗高、运算速度慢。主要用于军事和国防尖端技术领域,IBM 公司 1954 年 12 月推出的 650 机是第一代计算机的代表。

从 20 世纪 50 年代后期到 60 年代中期为第二代,即晶体管时期。计算机的主要逻辑元件由体积小、功耗低、价格便宜的晶体管所取代,存储器也改用速度更快的磁芯存储器。与此同时高级编程语言等软件陆续出现,提高了计算机的功能,并开始科研及商业中推广应用。这一时期主要代表有 DEC 公司 1957 年推出的 PDP 机、IBM 公司 1962 年推出的 7094 机、CDC 公司 1964 年研制成功的 6600 机等。CDC 公司 1969 年推出的晶体管计算机其运算速度已达每秒千万次浮点运算。

从 20 世纪 60 年代中期到 70 年代中期为第三代,即集成电路时期。由于采用了将逻辑元件与存储器件进行封装技术,取代了第一和第二代分离器件,从而提高了运算速度,降低了功耗和成本,体积也大大缩小,各部分的互连更加简单、可靠。与此同时,多种程序设计语言日趋成熟,开始了大型操作系统的研制,计算机的整体性能大幅度提高,拓宽了在许多领域的应用。这一时期的典型代表有 IBM 的 System/360 和 DEC 的 PDP-8 等。

从 20 世纪 70 年代中期到 70 年代后期为第四代,即大规模集成电路(VLSI)时期。逻辑元件集成度超过 100 个门电路,每平方英寸包含 5 万个以上元件,存储器用 MOS 和双极性元件。计算机产品明显区分为小型机、中型机和大型机,它们都配有较完善的操作系统及语言处理程序软件。典型产品如 IBM4300 系列、DEC PDP-11 等。

实际上从第三代计算机出现之后,由于微型计算机的异军突起,硬件技术的飞速发展和软件系统层出不穷,已很难找一个统一的标准进行第几代的划分了。高档微机、高性能的工作站和服务器不断涌现,特别是近些年来,硬件技术的发展以每 18 个月集成度提高一倍的速度在发展(摩尔定律),为开发大规模、高复杂度的软件系统提供了宽广的舞台。

从 20 世纪 80 年代开始,陆续提出了第五代计算机研制计划,简称为 FGCS(Five Generation Computer System),其主要特征是不以硬件技术特性来区分,而更强调软件的性能与作用;虽然其关键技术仍是 VLSI 结构,但扩充了并行处理、基于关系数据库的知识库,以及人工智能和模式处理的应用;其主要目标是具有问题求解能力和智能接口等。因此,第五代计算机是具有人工智能知识信息处理和非数值计算能力的高性能计算机。

### 1.1.2 计算机软、硬件构成

1945 年冯·诺依曼提出了“存储程序”的工作原理,他与同事们合作研制的存储程序计算机 IAS 被称为冯·诺依曼机。如图 1-1 所示,其中主存用以存放程序和数据,算术逻辑运算单元 ALU 完成二进制算术和逻辑运算,程序控制单元 PCU 负责解释和执行指令,输入/输出设备由 PCU 控制,实现与外部环境的数据交换。

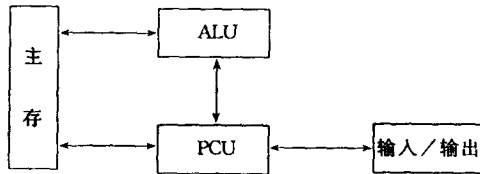
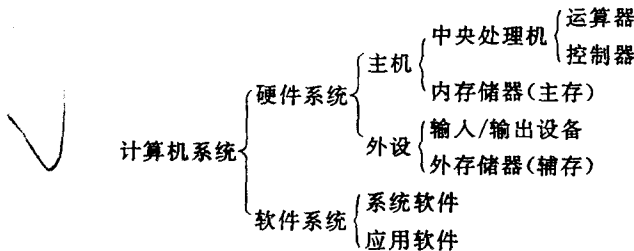


图1-1 IAS计算机结构

冯·诺依曼机奠定其后几乎所有通用计算机结构的基础,当今差不多各种通用型计算机都是冯·诺依曼机的扩充和完善。计算机系统是由硬件系统与软件系统构成的,也可称为计算机硬件与软件资源。详细分类如下:



这种分类表示在一般计算机教科书中,尤其是讲述硬件原理的书籍中是无可非议的。但不难发现,这种传统的分类恰恰反映了人们早期对计算机的认识,即软件作用还不是那么重要以及软件对硬件的过于依赖。实际上,通常将仅由硬件构成的、未配有任何软件的计算机称为裸机。若用户直接使用裸机,则只能利用提供的机器指令进行操作,这是十分困难的。一个计算机系统要能正常工作,并且充分发挥其硬件的各种功能,必须配备完善的软件系统,计算机系统的软件层次及其与硬件的关系见图 1-2。经过四五十年年的发展,软件系统也形成了庞大的家族,同样也可有它们的分类描述:

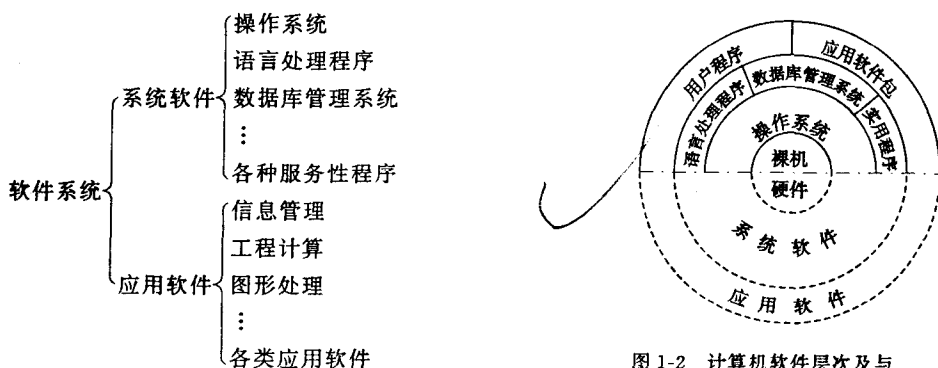


图 1-2 计算机软件层次及与硬件关系

其中系统软件一般指由计算机设计制造者提供给用户的、为方便用户使用和充分发挥计算机整体效能的系统程序。最基本的系统软件是操作系统,它负责管理和控制计算机的硬件资源和软件资源,是用户使用计算机的支持环境和操作平台,如 DOS、Windows、UNIX 和 Linux 等;语言处理程序是指把用非机器语言编写的程序翻译成可执行的机器指令的各种编译和解释程序,如 FORTRAN 编译程序和 BASIC 解释程序等;数据库管理系统是实现大量商用数据进行集中统一管理的软件,如 FoxBASE、Oracle、Sybase 等;服务性程序是帮助用户维护使用计算机、提供服务性手段而编制的各种实用程序,如杀病毒程序、压缩与解压缩程序等。应用软件是工作在系统软件之上、利用计算机所提供的各种系统软件为解决各类实际问题而编制的应用程序,体现了特定应用环境下的特定功能。如管理信息系统的事务处理和决策分析软件;科学研究和工程技术中的数理统计、工程计算、计算机辅助设计(CAD)软件;人们学习与生活中的计算机辅助教学(CAI)课件和多媒体应用及游戏软件等。总之,随着计算机深入到国民经济和社会生活的各个方面,各类应用软件更加层出不穷。

计算机硬件技术的发展推动了软件技术的进步,而软件技术的发展与完善又促进了硬件技术的进步,两者既相互依存,又互相影响。在计算机系统中,人们为了充分发挥软、硬件资源作用而获取整体性能的提高,已经打破软、硬件之间的严格界限,如通过软件实现内、外存结合的虚拟存储技术;用专用的芯片固化复杂的图像处理程序等。同样,应用软件与系统软件的界限也变得不那么清晰,不少应用软件随着它的完善和更广的适用性而逐渐变成某些应用领域的通用性软件,并随计算机硬件一起出售,从而演变为系统软件,如当前不少的工具软件就属于这种情形。

### 1.1.3 计算机软件定义

在 20 世纪五六十年代,人们一直认为软件就是程序。当时的软件主要包括语言处理程序、支持系统运行管理的服务程序及一般用户用高级语言编写的应用程序等。这时的软件概念是在基于手工方式进行程序设计的背景下形成的,从设计、编程到调试运行一般是由一个人独立完成,体现的是个人的设计风格和算法技巧。但 60 年代末期,在设计一些大型软件时出现了“危机”,突出的实例是 IBM 公司在研制 OS/360 时陷入失败的困境。其原因是从总体设计、各类人员配合、程序之间接口、直到调试完成,都面临需要统一的工程

软件=程序+文档

规范进行协调和组织的问题。因而在 1968 年北大西洋公司组织 (NTO) 举办的软件学术会议上,人们第一次提出“软件工程”的概念,并在 70 年代提出:“软件=程序+文档”的观点。这些提法既强调了文档在软件设计中的重要性,指出文档是软件的组成部分;同时,面对软件复杂性日益增长、开发周期和维护工作量急剧增大、软件的成本不断提高等问题,人们在考虑要像实施其他工程项目一样来对待软件研制的全过程。1983 年国际电子电气工程师协会 (IEEE) 组织明确给软件下了一个定义,即软件应包括计算机程序、实现此程序功能所采取的方法、规则以及与其相关的文档和在机器上运行它所需要的数据等。从此,传统的程序设计的观念被软件开发所代替,产生了一门从技术和管理两个方面研究更好地开发和维护计算机软件的新学科——软件工程。

软件工程强调采用工程的概念、原理、技术和方法指导软件的开发和维护。在软件研制开发过程中,若能严格遵循软件工程的方法论,便可提高软件开发的成功率,减少开发和维护中出现的问题。随着软件概念的更新和计算机技术的发展,软件产业应运而生,软件也逐渐成为一种独立的具有知识产权的产品。与此同时,从技术上人们开始致力去寻找一种新的软件开发模式,来摆脱软件开发的落后状况,实现软件研制和维护的自动化,这就是后来提出的软件支持环境的基本思想。所谓软件支持环境,一般是指在宿主硬件和宿主软件的基础上,用于辅助、支援其他软件的研制和维护的一组工具性软件。大型软件的研制和维护是一种费用大、耗时多的工程,功能强大的支持工具对于改进软件质量、提高软件生产率是非常重要的。近些年来,用于辅助式支持计算机软件的开发、运行、维护、模拟、移植或管理而研制的各种软件工具大量涌现,尤其是面向对象的技术和具有二进制代码软件可重用的组件技术,为软件支持环境发展注入新的活力,必将促进软件技术的不断发展。

## 1.2 程序设计语言

### 1.2.1 程序设计语言的分类

程序设计语言是指用来编写计算机程序的语言,是用户与计算机之间交流信息所用的一种工具,通常分为机器语言、汇编语言、高级语言、第四代语言 (4GL),以及面向对象语言等。

#### 1. 机器语言

这种语言是最早用于直接被计算机硬件识别和执行的语言。机器语言程序是用二进制的 0、1 代码编写的机器指令的代码序列。其优点是计算机能够直接执行,运行效率高;缺点是难懂、易出错,因机器不同而机器语言也不同,不能移植,没有通用性。

## 2. 汇编语言

此种语言是为了克服机器语言的弊病而设计的语言。其特点是用英文单词或缩写符号做助记符来表示机器指令,例如用 LOD 表示取数,ADD 表示加法,SUB 表示减法等。汇编语言比较直观,易于阅读和记忆。汇编语言中的语句与机器语言代码基本是一一对应的,它的书写格式在很大程度上取决于特定的计算机机器指令。

机器语言和汇编语言都是面向机器的程序语言,称之为低级语言,对于人们的抽象思维和交流是十分不便的。

## 3. 高级语言

所谓“高级”主要是它采用了与人们的自然语言(主要指英语)相近的词汇和数学表达形式,使程序编写更接近人们处理问题的习惯方式,能适合描述解决各种问题的算法。高级语言使程序设计脱离了具体的计算机结构,不必了解机器的指令系统,从而具有通用性和使用的方便性,很容易推广普及。传统的高级语言一般是面向过程的程序语言,程序设计工作主要包括数据结构和算法设计。它以问题的过程处理为中心,其中数据结构用来量化描述需要解决的问题,算法设计则研究如何用更快捷、高效的方法来组织解决问题的具体过程。高级语言发展了结构化程序设计思想,对程序结构作了改进,使程序由顺序、选择和循环三种基本结构组成;对于复杂问题采用模块化、自顶向下、逐步求精的设计原则,因此程序结构清楚、易读、容易维护。

## 4. 第四代语言

这一提法主要流行在面向对象的程序设计方法尚未成熟的时期,4GL 主要指数据库及应用开发语言。这种语言的特点是用户只需要告诉计算机做什么,但不必告诉它怎么做,计算机就会自动地完成所需的操作,如报表或屏幕生成语言。4GL 是高度非过程化的语言。

## 5. 面向对象的语言

它继承了面向过程的高级语言的结构化程序设计、模块化、并行处理等优点,克服了数据与代码分离的弊端。它代表了新颖的程序设计思维方法,将人们研究的任何事物视为对象,用语言中的对象和类直接模拟现实世界中的事物,使解决问题的过程更符合人们的思考方式。面向对象语言将事物状况和操作封装于对象体中,类是同种对象的集合与抽象,类与类之间可以组成继承层次,并规定它们之间传递消息的规则。从程序的执行看,它归结为各对象以消息传递方式进行的通信;子类可以继承父类中的属性和操作,也可以定义自己的属性和操作,从而使内部表示有差异的对象可以共享与它们结构有相同部分的有关操作,达到代码重用的目的。

## 1.2.2 语言处理程序

语言处理程序主要包括汇编程序、解释程序和编译程序。

### 1. 汇编程序

它是针对汇编语言的翻译程序。人们用汇编语言编写的程序称为汇编语言源程序,这种源程序必须经过转换翻译成机器语言程序才能被计算机识别和执行。担当转换翻译任务的程序就是汇编程序,经过汇编程序翻译出来的机器语言程序称为目标程序。汇编程序的执行过程见图 1-3。

汇编程序在加工源程序时总是从头到尾地对源程序中的符号一个一个地阅读分析,这种运作称为扫描。一般用两遍扫描来完成对源程序的转换。第一遍把源程序中出现的名字进行造表,确定每个名字将占用内存位置;第二遍扫描时,按所造出的表把每条原为符号化的机器语言代换(换码)成二进制数码形式的机器指令。另外,汇编程序还必须具备一些附加的功能,例如查错功能、修改功能、打印功能和执行伪指令功能等。此外还有反汇编程序,用来把机器语言程序转换为汇编语言程序,可用于检查目标程序的正确性,也可作为一种“破译”程序的工具软件。

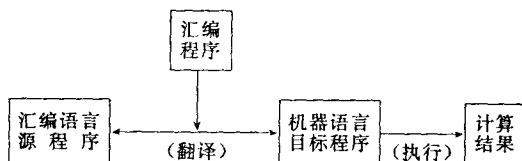


图1-3 汇编程序执行过程

### 2. 解释程序

它是高级语言翻译程序的一种。它的工作方式是用某种高级语言编写的源程序作为输入,将源程序的各种符号照原样或加以若干变形处理后,存放在存储器中,执行时再按顺序取出各结构单位,解释一句执行一句,其执行过程见图 1-4(a)。由于是对源程序边解释边执行,因此执行速度要慢一些,不适合处理大型计算问题。但是,解释程序的这种工作方式,对语言一级的源程序进行变更、检查和修改比较方便,并一般具有人机会话的功能。因此,虽然解释程序处理的效率不高,但它还是被有效地使用。

### 3. 编译程序

它是一种将用高级语言编写的源程序翻译成机器语言或汇编语言之类的“低级语言”的目标程序的实用程序。这种目标程序可能是汇编语言形式,还需经汇编程序翻译后才能执行。其执行过程见图 1-4(b)。由于高级语言与计算机的关系不紧密,不具有与机器语言直接的对应关系,因此,对这种语言的处理过程是相当复杂的。编译程序执行中一般分成词法分析、句法分析和语义分析、代码优化及生成、符号表管理等部分,一条高级语言语句最后将转换成几条乃至几十条机器语言指令。在编译程序处理过程中,如果发现源程序描

述上有语法错误,将给出纠正程序错误的有效信息。



图1-4 解释程序和编译程序执行过程

### 1.2.3 典型的高级语言

高级程序设计语言至今已有几百种,较通用的也有几十种,各有其特点及应用领域,用途较广的有如下几种。

(1)BASIC 语言:是一种适宜初学者的高级语言,一般是解释执行的会话型语言。多年来有很大发展,版本较多,功能也不断增强。它简单易学,被广泛地应用在微机系统中。

(2)FORTRAN 语言:是第一个被广泛应用于科学计算的高级语言。一个 FORTRAN 程序由一个主程序或一个主程序和若干个子程序组成。程序的执行由主程序开始,主程序块可以调用子程序块,子程序块间也可以互相调用,但子程序块不能调用主程序块。程序中的变量地址是静态分配的,不能包含递归子程序和动态数组。FORTRAN 程序结构简单,可分块书写编译,计算精度高,使用起来灵活、方便。

(3)ALGOL 语言:是一个早期研制出来的高级语言。它具有严格的文法规则,具有分程序结构,整个 ALGOL 程序本身就是一个分程序。分程序结构是可以嵌套的,即分程序可含有别的分程序,过程也可以看成是一个分程序,这个分程序可以被别的分程序调用。变量地址是动态分配的,允许递归子程序和动态数组。ALGOL 程序结构严谨、清晰,早期应用较广。

(4)COBOL 语言:是面向商用事务处理的高级语言。它引入了文件和记录的逻辑概念,按层次结构描述数据,在字符串运算和编辑上首次扩充非数值运算领域的数据处理能力。COBOL 程序语言采用了更接近自然语言的形式,被广泛应用于商业及行政管理部

门。

(5)PASCAL 语言:是一种典型的结构化程序设计语言。它扩充了数据结构功能,允许用户定义数据类型,将分程序和过程概念统一为过程,一个 PASCAL 程序可看成是一个由操作系统所调用的过程。PASCAL 过程是可以嵌套和递归的,还增加了并行控制结构。PASCAL 虽然问世较晚,但得到人们极大的重视,在计算机科学的教科书中多采用这种结构严谨、描述简明的语言。

(6)C 语言:是 20 世纪 70 年代发展起来的通用性程序设计语言,1972 年首先在 PDP-11 机器的 UNIX 操作系统上实现的。C 语言的开发深受 BCPL 和 B 等语言的影响,UNIX 操作系统主体就是用 C 语言书写的,但又独立于 UNIX 系统。它将字符、整数和浮

点数当作基本的数据对象,但可用指针、数组、结构和联合建立新的数据类型。C 语言提供编写结构良好的程序所需要的控制结构:语句复合;条件选择(if);在开头测试终止条件(While for)和在结束处测试终止条件(do)的循环;分情形选择(case)等。由于它具有指针和地址操作的能力,因此说 C 语言还是一种面向机器的低级语言。C 语言吸取多种语言的先进控制结构和数据描述的特性,具有简单灵活、易于实现、方便移植等优点,因而成为最具生命力的语言。它不仅用来书写操作系统、编译程序、数据库管理系统等系统软件,还可以用来开发管理信息系统、工业过程自动控制、图形和图像处理等各种应用软件。

### 1.2.4 面向对象的语言

面向对象的方法早在 20 世纪 60 年代就在实验室中提出了。最早的面向对象的软件是 1966 年开发的 Simula I,这是一种用于模拟(或称仿真)的语言,但它采用 ALGOL 60 分程序概念,使用了对象、类及继承的术语。直到 Simula 67,面向对象的语言仍处在酝酿之中。第一个称得上面向对象语言是 Smalltalk,这个名字取自“少说话(talksmall)”,意思是你通过很少话语就可完成很多工作。商品化的 Smalltalk 强调对象概念的统一,并引入完善的类、方法、实例等概念和术语,应用了单重继承和动态链接,已全面具备了面向对象语言的特征。其后又产生了 Lisp、Object 及面向对象的 PASCAL 等,但近年来最有影响并对面向对象技术的普及和推动起最大作用的当属 C++ 和 Java。

C++ 语言在兼容原有最流行的 C 语言基础之上,加上了面向对象的有关内容和规则。由于它的很多语法规则与 C 语言相近,所以很容易为广大的 C 程序员所接受;同时 C++ 所具有的面向对象的功能可简化应用软件的开发、设计和维护,为大型软件开发提供了极大方便,C++ 已正在逐渐取代 C 而成为主流的编程语言。C++ 的广泛推广和成功应用证明了面向对象技术的实力和发展前景。

Java 语言是 20 世纪 90 年代出现的面向对象的编程语言。相对于 C++,Java 去掉了 C++ 为了兼容 C 语言而保留的非面向对象的内容,使程序更加严谨、可靠、易懂。尤其是 Java 所特有的“一次编写、多次使用”的跨平台的优势,使得它将特别适合在网络应用开发中使用,成为面向对象开发工具中极具潜力的新秀。

## 1.3 网络基础与应用

计算机网络是计算机技术与通信技术日益发展和密切结合的产物,正成为信息化社会十分重要的基础设施。我们可以将计算机网络定义为:采用通信手段,将地理位置分散的、各自具备自主功能的若干台计算机有机地连接起来的复合系统,这个复合系统可用来实现通信交往、资源共享或协同工作等目标。

### 1.3.1 计算机网络的发展

第一代计算机网络实际上是指以单台计算机为中心的远程联机系统。最早追溯到 20



世纪 50 年代中叶,美国的半自动地面防空系统(SAGE)就开始进行计算机与通信技术相结合的尝试,把远距离雷达和其他测控设备的信息经由通信线路汇集到一台 IBM 计算机上进行集中处理与控制;60 年代初美国航空公司的机票预订系统 SABREI,由一台中心计算机和全美范围内 2000 多个终端相联,这一时期的网络也称面向终端的计算机网络。

第二代计算机网络是多个主计算机通过通信线路连接起来,为用户提供服务的系统,这是 60 年代后期兴起的。美国国防部高级研究计划局给许多大学和公司提供经费,以促进多个主计算机互联网的研究,简称 ARPA 网。它把一个计算机网络划分成通信子网和资源子网两大部分,见图 1-5。

图中 H 为主机(Host),IMP 为通信节点(接口报文处理机)。各主机都具有自主处理能力,通过通信子网负责运行用户程序,向网络用户提供共享的软硬件资源,它们组成资源子网。IMP 和它们之间互连的通信线路一起负责主机之间的通信任务,构成通信子网,IMP 之间采取存储转发方式进行报文的传递。当今网络仍在沿用这种组成方式。

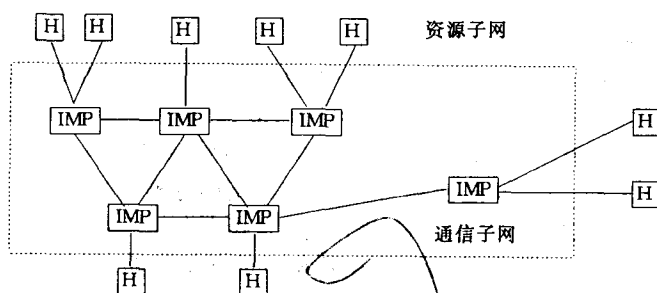


图1-5 计算机网子网构成

第三代计算机网络是开放式和标准化网络。由于第二代网络大都由大公司、研究单位和大学各自研制的,没有统一的网络体系结构,如 IBM 有 SNA 网,DEC 推出 DECNET 网等,均是各自厂商同种计算机互连的封闭系统,要把不同厂商的计算机和网络互连在一起是极其困难的。70 年代后期人们认识到第二代网络的不足。经过若干年卓有成效的工作,国际标准化组织(ISO)1984 年正式颁布了一个称为“开放系统互连基本参考模型”(简称 OSI 模型)的国际标准。在开放网络中,只要网络中的计算机系统和通信设备都能遵守共同认可的国际标准,就能保证顺利地相互通信,并允许方便地在构建网络时增加和删去符合标准的各类计算机系统和网络设备。OSI 模型目前已被国际社会普遍接受,并公认为计算机网络体系结构的基础。各大计算机公司都相继宣布支持 OSI,并争相开始研制符合 OSI 标准的软硬件产品。同时,以 ARPA 为骨干的国际互联网 Internet(简称因特网,互联网)逐步演变和发展起来。它是将成千上万个各种计算机网络互连起来的网际网,但对用户看来就如同是一个计算机网络一样。因特网也是一个开放式和标准化的网络,它的体系结构和 OSI 模型基本相吻合。它所遵循的一系列的标准协议中,以著名的传输控制协议(简称 TCP)和网际协议(简称 IP)为核心,虽然它不是完全的 OSI 模型的标准,但与其有密切对应关系,并且由于其被广泛地采用,已成为事实上的国际工业标准。

第四代计算机网络也可称为新一代计算机网络,是目前正在探讨和发展中的网络结构。因为因特网的发展面临着带宽(即网络传输速率与流量)的限制、网上安全管理、多媒